

Servidor Web com Node.js e Express

Objetivo: Neste guia, vamos explorar o que é o Node.js e como usá-lo para criar um pequeno servidor web. Ao final, você terá um site local rodando no seu computador, exibindo uma página HTML criada e personalizada por você.

Antes de Começar: Uma Pequena Analogia

Imagine que você quer construir uma cozinha de restaurante do zero.

- **Node.js é a própria cozinha:** Ele te dá o espaço, a eletricidade, o gás e a água. É um ambiente poderoso que permite que as coisas funcionem, mas ele não te diz *como* cozinhar ou organizar os pedidos. Ele é a base de tudo.
- **Express.js é o Chef de Cozinha experiente:** Você contrata um Chef (Express) para organizar essa cozinha. Ele cria estações de trabalho, define receitas fáceis de seguir e gerencia como os pedidos dos clientes (requisições da web) são recebidos, preparados e entregues na mesa (seu navegador).

Em resumo, o **Node.js** é o ambiente que executa nosso código no servidor, e o **Express** é uma ferramenta (um framework) que torna a tarefa de construir esse servidor muito mais simples e organizada.

Parte 1: Preparando o Ambiente (Instalando o Node.js)

Nosso primeiro passo é construir a "cozinha". Para isso, precisamos instalar o Node.js na sua máquina.

1. **Ferramenta de Download (`curl`):** Antes de prosseguir, precisamos verificar se você tem uma ferramenta de linha de comando chamada `curl`. Ela é essencial para baixar arquivos e scripts da internet diretamente pelo terminal. Caso não a tenha, **use o gerenciador de pacotes do seu sistema operacional (como o `apt` no Ubuntu) para instalá-la.**
2. **Baixando o Script de Instalação:** Agora, vamos usar o `curl` para baixar um script especial. Este script não instala o Node.js diretamente, mas ele configura seu sistema para que ele saiba exatamente de onde baixar a versão correta (v23.x).

```
curl -fsSL https://deb.nodesource.com/setup_23.x -o nodesource_setup.sh
```

3. **Executando o Script de Configuração:** Com o script em mãos, precisamos executá-lo com permissões de administrador. Isso irá preparar seu gerenciador de pacotes para a próxima etapa.

```
sudo -E bash nodesource_setup.sh
```

4. **Instalando o Node.js:** Com tudo preparado, agora sim, **use o gerenciador de pacotes do seu sistema (`apt`) para instalar o pacote `nodejs`**. Ele agora saberá buscar a versão que configuramos.
5. **Verificação Final:** Para ter certeza de que a "cozinha" foi montada corretamente, peça para o Node.js dizer qual é a sua versão. Se um número de versão aparecer, significa que a instalação foi um sucesso!

```
node -v
```

Parte 2: Estruturando o Nosso Projeto

Com o Node.js instalado, vamos organizar nosso espaço de trabalho.

1. **Crie a Pasta do Projeto:** Crie um diretório (uma pasta) em seu computador para este projeto. Dê a ele um nome claro, como `meu-primeiro-servidor`, e navegue para dentro dele pelo terminal.
2. **Iniciando o Projeto com NPM:** O Node.js vem com um assistente chamado **NPM** (Node Package Manager). Pense nele como o gerente do nosso restaurante. Ele controla todos os "ingredientes" (pacotes de código) que nosso projeto precisa. Para começar, diga ao NPM para criar um "cartão de identidade" para o nosso projeto. Este cartão é um arquivo chamado `package.json`.

```
npm init -y
```

O `-y` apenas diz "sim" para todas as perguntas padrão, criando um arquivo básico.

3. **Contratando o Chef (Instalando o Express):** Agora, vamos usar o NPM para "contratar" nosso Chef, o Express. Este comando baixa o Express e o salva em uma pasta especial chamada `node_modules` (o almoxarifado de ferramentas do nosso projeto).

```
npm install express
```

4. **A Área Pública do Site:** Em um servidor web, precisamos de um lugar para colocar os arquivos que os visitantes podem ver, como páginas HTML, imagens e CSS. Por convenção, criamos uma pasta chamada `public` para isso. Crie esta pasta dentro do diretório do seu projeto.

Parte 3: Criando a Página Principal

Vamos criar a página que será exibida aos visitantes.

1. **Crie o Arquivo HTML:** Dentro da pasta `public` que você acabou de criar, crie um novo arquivo chamado `index.html`.
2. **Edite sua Página:** Abra o arquivo `index.html` com seu editor de texto ou IDE favorito. Cole o código abaixo. Esta é a estrutura básica da sua página.

Atenção: Modifique as linhas indicadas, substituindo o texto genérico pelo seu nome e sua turma!

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Meu Site Pessoal</title>
  <style>
    body { font-family: Arial, sans-serif; background-color: #e9e9e9; display: flex; justify-content: center; }
    .card { background-color: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
    h1 { color: #2c3e50; }
    p { font-size: 18px; color: #34495e; }
  </style>
</head>
<body>
  <div class="card">
    <h1>Meu Primeiro Servidor Web!</h1>
    <p><strong>Criado por:</strong> SEU NOME COMPLETO AQUI</p>
    <p><strong>Turma:</strong> SUA TURMA AQUI</p>
  </div>
</body>
</html>
```

Parte 4: Escrevendo as Instruções para o Servidor

Agora, vamos criar o cérebro do nosso servidor. Este arquivo dará as instruções para o Node.js e o Express.

1. **Crie o Arquivo do Servidor:** Na pasta principal do seu projeto (ao lado da pasta `public`), crie um arquivo chamado `server.js`.
2. **Programa o Servidor:** Abra o arquivo `server.js` e cole o código a seguir. Leia os comentários para entender o que cada linha faz.

```
// 1. "Contratamos" o chef Express que instalamos anteriormente.
const express = require('express');

// 2. Criamos nossa aplicação, que será o nosso servidor.
const app = express();

// 3. Definimos um número de "porta" para o nosso restaurante.
// Pense nisso como o número do apartamento onde nosso servidor vai morar.
const PORT = 3000;

// 4. Esta é a instrução mais importante!
// Estamos dizendo ao Chef (Express): "Qualquer arquivo que um cliente pedir,
// primeiro procure dentro da nossa pasta 'public'".
app.use(express.static('public'));

// 5. Ligamos o servidor e o colocamos para "ouvir" na porta definida.
// A mensagem no console nos avisará quando tudo estiver pronto.
app.listen(PORT, () => {
  console.log(`✅ Servidor iniciado e ouvindo na porta ${PORT}`);
  console.log(`Acesse o site em: http://localhost:${PORT}`);
});
```

Parte 5: Inaugurando o Servidor!

Tudo está pronto para a grande inauguração.

1. **Execute o Servidor:** No seu terminal, certifique-se de que você está na pasta principal do projeto e execute o seguinte comando:

```
node server.js
```

2. **Visite seu Site:** Se tudo deu certo, você verá as mensagens de confirmação no terminal. Agora, abra seu navegador de internet e vá para o endereço: **<http://localhost:3000>**